

A The dependency graph 401 forms a network of dependencies between the nodes. For example, the leaf node e 410 corresponds to a scalar property e, and the expression node c 408 corresponds to the variable or dynamic property c. Expression node c 408 depends on the leaf node e 410 and is its dependent node. Similarly, the leaf node e 410 is a dependency of the expression node c 408. Because the dependency is direct, expression node c 408 is a least-dependent node.

The paragraph beginning on page 14, line 9, has been replaced with the following rewritten paragraph:

A2 The recalc engine 308 also creates a variable table 400, which is a lookup table of object/dispid pairs. In one possible embodiment, the lookup table is formed from a hash table of IDispatch objects. Each object entry is a list of dispid entries that correspond to a particular variable and point to the actual nodes in the dependency graph 401. The variable table 400 provides quick mapping between a variable and a node in the dependency graph 401. The variable table 400 also sets up and receives event notifications, which allows the object to listen for or notice events that affect a change in a property value. One possible mechanism for setting up and receiving event notifications is the OLE interface, IPropertyNotifySink.

C. In the Claims

Please amend claims 5, 6, 13, 17, and 24 as follows:

A3 5. (Once Amended) A computer-readable medium having stored thereon a data structure, the data structure including a plurality of nodes, comprising:

- at least one leaf node stored in memory, each leaf node containing a scalar property;
- at least one expression node stored in memory, each expression node containing an expression written in a markup language and defining a dynamic property, the expression being a function of the scalar property;
- at least one pointer stored in memory, each pointer mapping a dependent/dependency relationship between the scalar properties and the expressions, wherein upon

A3
notification of a change in the value for one of the scalar properties, a browser executes the expressions dependent on the scalar property having a changed value.

6. (Once Amended) The computer-readable medium of claim 5 wherein the data structure is formed by a browser.

13. (Once Amended) The computer-readable medium of claim 12 wherein:

two pointers map the second expression to the first expression and a third expression, respectively, the second expression being dependent on the first and third expressions thereby being more dependent than the first and third expressions; and

the browser does not execute the second expression until the first and third expressions are executed.

17. (Once Amended) The method of claim 16 wherein:

the dependency graph includes a plurality of expression nodes mapped in network wherein the most-dependent expression node in the network has no dependent expression nodes and the least-dependent expression node depends directly on a scalar property; and

recalculating the values of expressions stored in nodes having a dirty state includes (a) executing the expression corresponding to each of the expression nodes in the network beginning with the least-dependent node and ending with the most-dependent node, and (b) assigning the value of each executed expression to the dynamic property corresponding to that executed expression.